



Penerapan Algoritma *Greedy* Pada Mesin Penjual Otomatis (*Vending Machine*)

Alamsyah¹, Indriani Tiara Putri²

¹Jurusan Ilmu Komputer, FMIPA, Universitas Negeri Semarang

²Jurusan Matematika, FMIPA, Universitas Negeri Semarang

Email: ¹alamsyah@mail.unnes.ac.id, ²tputri@gmail.com

Abstrak

Dalam memasarkan produk minuman dan makanan ringan, Indonesia masih banyak menggunakan tenaga manusia untuk menyalurkan produk tersebut dari pabrik sampai ke konsumen akhir. Jika setiap toko membeli produk dalam jumlah banyak, maka didapatkan keuntungan yang besar dalam penjualan produk tersebut. Hal ini menyebabkan harga penjualan produk yang sampai ke konsumen akhir lebih mahal daripada harga asli yang diberikan oleh pabrik. Dengan permasalahan tersebut, penulis mencoba membuat aplikasi mesin penjual otomatis (*vending machine*). Pada umumnya *vending machine* tidak memberikan uang kembalian. Disini penulis mencoba membuat *vending machine* dengan menerapkan algoritma *Greedy* agar dapat memberikan uang kembalian sehingga harga penjualan produk sesuai dengan harga asli pabrik. Algoritma *Greedy* diterapkan untuk menentukan pecahan berapa saja yang muncul dalam proses pengembalian uang dengan meminimalkan jumlah uang logamnya. Penulis menggunakan aplikasi *Visual Basic* untuk menerapkan program *vending machine*.

Kata Kunci: *Vending machine*, Algoritma *greedy*, *Visual basic*

1. PENDAHULUAN

Indonesia masih menggunakan tenaga manusia untuk memasarkan produk minuman dan makanan ringan dari pabrik sampai ke konsumen akhir. Dengan cara pemasaran seperti itu akan diketahui berapa keuntungan yang diperoleh setiap toko dalam penjualan produk tersebut. Jika setiap toko membeli produk dalam jumlah banyak, maka toko tersebut akan memperoleh potongan harga dari pabrik. Secara otomatis jika setiap toko menjual produk tersebut dengan memperoleh keuntungan, maka keuntungan yang diperoleh setiap toko relatif besar. Hal ini menyebabkan harga jual yang diberikan oleh setiap toko pada konsumen akhir akan lebih mahal dibandingkan dengan harga asli dari pabrik. Selain itu harga jual setiap toko akan berbeda – beda tergantung toko tersebut menjadi urutan ke berapa dalam urutan pendistribusian produk dari pabrik.

Dengan perkembangan ilmu pengetahuan dan teknologi, penulis merancang sebuah program otomatis di bidang penjualan yaitu *vending machine*. *Vending machine* ini memang masih jarang ditemukan di Indonesia, akan tetapi sudah banyak ditemukan di negara maju lainnya. Dengan adanya *vending machine* konsumen akan dilayani secara otomatis tanpa harus ada perantara manusia dan harga jual produk yang diperoleh

konsumen sesuai dengan harga asli pabrik. *Vending machine* akan bekerja selama 24 jam. Jadi konsumen dapat membeli produk yang diinginkan kapan saja. Selain itu, cara kerja *vending machine* sangat cepat dan efisien sehingga konsumen lebih cepat memperoleh produk yang diinginkan tanpa harus mengantri lama [1], [2]. Program *vending machine* dapat menerima masukan uang kertas maupun uang logam, tetapi uang kembalian yang diperoleh konsumen hanya berupa uang logam.

Penulis akan membahas tentang bagaimana *vending machine* memberikan uang kembalian secara optimal, yaitu meminimalkan keluarnya uang logam dalam proses pengembalian uang. Program *vending machine* ini akan menerapkan algoritma *Greedy* untuk menentukan pecahan berapa saja yang akan muncul dengan meminimalkan munculnya uang logam saat proses pengembalian uang.

2. METODOLOGI

2.1. Tahapan Perancangan

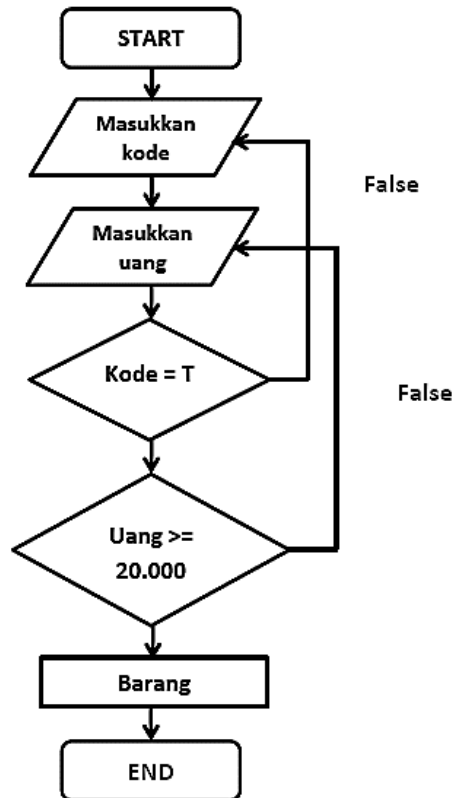
Pada tahap perancangan, terlebih dahulu dipahami bagaimana sistem kerja *vending machine*. *Vending machine* akan bekerja untuk mengeluarkan barang jika dimasukkan uang kertas maupun uang logam terlebih dahulu. Program *vending machine* dirancang sedemikian rupa agar dapat memberikan uang kembalian berupa uang logam. Dengan menerapkan algoritma *Greedy* pada program *vending machine* [3], maka uang kembalian yang diterima konsumen akan diminimalkan jumlah uang logamnya.

Langkah-langkah dalam perancangan sistem sebagai berikut. (a) Pendefinisian kebutuhan, meliputi penentuan sasaran atau objek untuk program yang akan dibuat. Pada *vending machine* yang dibuat, konsumen akan menerima uang kembalian berupa uang logam sehingga harga jual barang sesuai dengan harga yang diberikan pabrik, (b) Perancangan perangkat lunak meliputi, memasukkan uang (*Vending machine* dapat menerima uang masukan berupa uang kertas maupun uang logam. Nominal uang kertasnya adalah Rp 2.000,00; Rp 5.000,00; dan Rp 10.000,00. Sedangkan nominal uang logamnya adalah Rp 100,00; Rp 200,00; Rp 500,00 dan Rp 1.000,00), memasukkan kode minuman atau makanan, minuman atau makanan keluar sesuai dengan yang dipilih konsumen, Apabila uang yang dimasukkan jumlahnya lebih besar dari harga minuman atau makanan pada *vending machine*, maka uang kembalian berupa uang logam akan muncul; (c) Penerapan rancangan, merupakan proses pengkodean rancangan yang telah dibuat ke dalam *Visual Basic* untuk membuat program *vending machine* yang menerapkan algoritma *Greedy*; (d) Setelah program selesai dibuat, maka program siap diuji untuk mengetahui apakah program tersebut berhasil atau gagal.

2.2. Flowchart dan Algoritma

Flowchart adalah sekumpulan simbol yang menunjukkan atau menggambarkan rangkaian kegiatan – kegiatan program dari awal sampai selesai pada suatu algoritma.

Flowchart untuk *vending machine* ditunjukkan pada Gambar 1.



Gambar 1. Flowchart untuk *vending machine*

Algoritma adalah metode efektif dari perintah–perintah yang telah didefinisikan dengan baik untuk menghitung sebuah fungsi. Dimulai dari sebuah kondisi awal dan *input* awal, intruksi–intruksi tersebut menjelaskan sebuah komputasi yang bila dieksekusi atau diproses lewat sejumlah urutan kondisi terbatas dapat terdefinisi dengan baik dan menghasilkan keluaran (*output*). Dalam hal ini digunakan algoritma *Greedy* untuk menyelesaikan masalah bertahap (langkah per langkah) dimana pada setiap langkah akan diambil pilihan terbaik (optimum lokal) dengan mengharapkan hasil akhirnya optimum global. Untuk skema umum algoritma *Greedy* seperti diperlihatkan pada Gambar 2.

```

function greedy (input C: himpunan_kandidat) → himpunan_kandidat
{ Mengembalikan solusi dari persoalan optimasi dengan algoritma greedy
  Masukan: himpunan_kandidat C
  Keluaran: himpunan solusi yang bertipe himpunan_kandidat }

Deklarasi
  x : kandidat
  S : himpunan_kandidat

Algoritma:
  S ← {} { inialisasi S dengan kosong }

  while (not SOLUSI(S)) and (C ≠ {}) do
    x ← SELEKSI(C)           { pilih sebuah kandidat dari C }
    C ← C - {x}             { elemen himpunan kandidat
                             berkurang satu }
    if LAYAK(S ∪ {x}) then
      S ← S ∪ {x}
    endif
  endwhile
  { SOLUSI(S) or C = {} }

  if SOLUSI(S) then
    return S
  else
    write('tidak ada solusi')
  endif

```

Gambar 2. Skema umum algoritma *greedy*

Untuk elemen–elemen algoritma *Greedy* pada *vending machine* adalah sebagai berikut.

(a) Himpunan kandidat, C. Himpunan kandidat berisi elemen–elemen pembentuk solusi permasalahan. Pada *vending machine* himpunan kandidatnya adalah himpunan uang logam dengan nilai Rp 100,00; Rp 200,00; Rp 500,00; dan Rp 1.000,00; dan uang kertas dengan nilai Rp 2.000,00; Rp 5.000,00; dan Rp 10.000,00, (b) Himpunan solusi, S. Himpunan solusi berisi kandidat–kandidat terpilih dari solusi permasalahan. Himpunan solusi pada *vending machine* yaitu memunculkan uang kembalian berupa uang logam yang jumlahnya diminimalkan, (c) Fungsi seleksi (*selection function*). Fungsi seleksi akan digunakan pada setiap langkah untuk memilih kandidat terbaik dimana kandidat yang sudah dipilih tidak akan dipertimbangkan pada langkah selanjutnya. Fungsi seleksi yang dilakukan *vending machine* yaitu dengan memilih nilai uang logam terbesar pada himpunan kandidatnya, (d) Fungsi kelayakan (*feasible*). Fungsi ini digunakan untuk memeriksa apakah suatu kandidat yang dipilih sudah memberikan solusi yang layak, yaitu kandidat tersebut bersama – sama himpunan solusi yang terbentuk tidak melanggar kendala yang ada. Apabila kandidat yang sudah

dipilih layak, maka dimasukkan pada kandidat solusi. Jika sebaliknya, maka kandidat tersebut dihapus atau dibuang. Pada *vending machine* fungsi ini akan memeriksa apakah uang kembalian yang dikeluarkan *vending machine* tidak melebihi jumlah uang yang harus diterima dan (e) Fungsi obyektif. Fungsi ini akan memaksimalkan atau meminimalkan solusi. Pada permasalahan *vending machine* fungsi ini akan meminimalkan keluarnya uang logam sebagai uang kembalian.

Skema algoritma *Greedy* untuk *vending machine* seperti diperlihatkan pada Gambar 3. Misalkan uang kembalian adalah R.

```
function PengembalianUang(input C : himpunan_uang_logam, R : integer)
→ himpunan_uang_logam
{ mengembalikan uang logam yang total nilainya = R, tetapi jumlah uang
  logamnya diminimalkan }

Deklarasi
x : uang_logam
S : himpunan_uang_logam

Algoritma
S ← {}

while (  $\sum(\text{nilai semua uang logam di dalam } S) \neq A$  ) and ( C ≠ {} ) do
  x ← uang_logam yang mempunyai nilai terbesar
  C ← C - {x}
  if (  $\sum(\text{nilai semua uang logam di dalam } S) + \text{nilai uang logam } x \leq$ 
    A then
    S ← S ∪ {x}
  endif
endwhile

if (  $\sum(\text{nilai semua uang logam di dalam } S) = A$  then
  return S
else
  write('tidak ada solusi')
endif
```

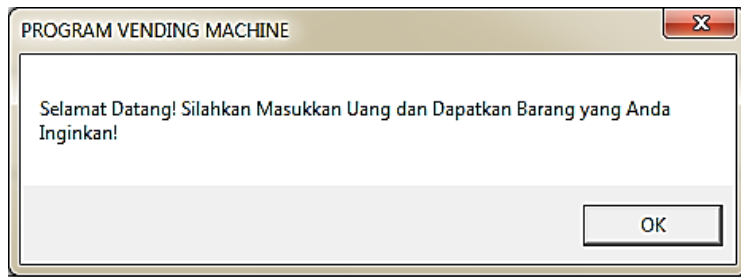
Gambar 3. Skema umum algoritma *greedy*

3. HASIL DAN PEMBAHASAN

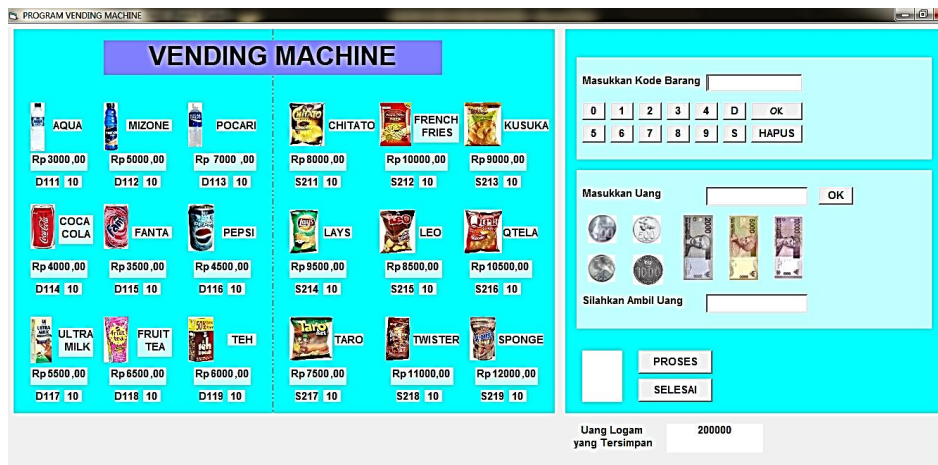
Vending machine (mesin penjual otomatis) adalah mesin yang dirancang untuk mengeluarkan barang yang diinginkan secara otomatis tanpa perantara manusia dengan cara memasukkan uang kertas maupun uang logam terlebih dahulu. Program *vending machine* yang disajikan memiliki kemiripan cara kerja sebagaimana cara kerja dengan

vending machine umumnya. Perbedaannya adalah pada uang pengembaliannya.. Uang kembalian yang diperoleh konsumen hanya berupa uang logam saja.

Dalam pembuatan sistem *vending machine* digunakan *Visual Basic* dengan menerapkan algoritma *Greedy* pada proses uang kembalian. Berikut ini tampilan program yang menggunakan *Visual Basic*. Tampilan awal saat program dijalankan adalah seperti diperlihatkan oleh Gambar 4. Setelah di jalankan maka tampilan halaman utama program *vending machine* ini seperti ditunjukkan pada Gambar 5.



Gambar 4. Tampilan awal saat program *vending machine* di run



Gambar 5. Tampilan halaman utama saat program *vending machine* di run

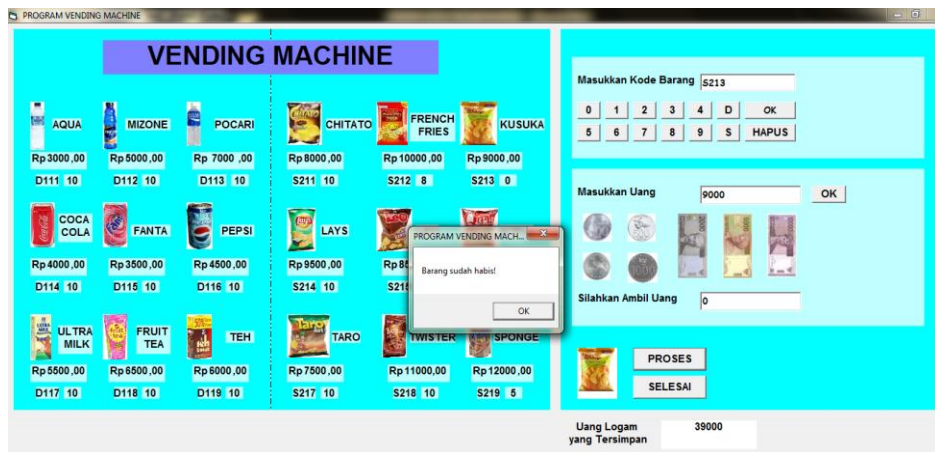
Apabila kita memasukkan kode minuman atau makanan yang diinginkan kemudian memasukkan uang sesuai atau lebih besar dari yang tertera pada *vending machine*, maka tampilannya akan seperti Gambar 6.

Penerapan Algoritma Greedy Pada Mesin Penjual Otomatis (*Vending Machine*)



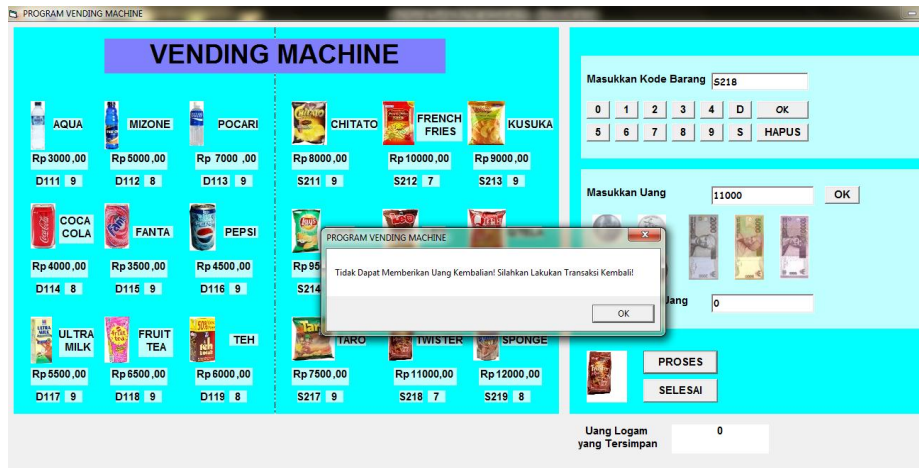
Gambar 6. Tampilan saat program *vending machine* dijalankan

Vending machine yang dibuat hanya dapat menyimpan minuman atau makanan masing-masing sebanyak 10 barang. Apabila stok barang pada *vending machine* tersebut habis, maka tampilannya akan seperti pada Gambar 7.



Gambar 7. Tampilan apabila barang sudah habis

Ruang penyimpanan uang logam yang dijadikan sebagai uang kembalian pada *vending machine* jumlahnya terbatas, yaitu sebesar Rp 200.000,00. Dan bila uang logam untuk kembalian tersebut habis maka akan muncul tampilan seperti Gambar 8.



Gambar 8. Tampilan apabila uang logam yang tersimpan habis

Dari Gambar 6 diketahui uang kembalian yang diperoleh adalah Rp 3.000,00. Dengan menerapkan algoritma *Greedy* akan diperoleh hasil uang kembalian sebagai berikut.

- Pada *vending machine* yang dibuat terdapat uang logam sebagai uang kembalian dengan nominal uang Rp 100,00; Rp 200,00; Rp 500,00; dan Rp 1.000,00.
- Pilih 1 buah uang logam yaitu Rp 1.000,00 sebagai uang terbesar yang masuk dalam fungsi kelayakan. Jadi uang kembalian yang tersisa Rp 2.000,00.
- Pilih 1 buah uang logam dengan nilai terbesar yaitu Rp 1.000,00. Jadi diperoleh uang Rp 1.000,00 sebanyak 2 dan uang kembalian bersisa Rp 1.000,00.
- Pilih uang logam dengan nilai terbesar lagi yaitu Rp 1.000,00. Jadi diperoleh uang Rp 1.000,00 sebanyak 3.

Jadi dengan uang kembalian sebesar Rp 3.000,00 akan diperoleh kembalian uang logam dengan uang Rp 1.000,00 sebanyak 3 uang logam.

Permasalahan lain misalnya uang kembalian yang muncul sebesar Rp 3.800,00. Dengan langkah yang sama seperti di atas, akan diperoleh data sebagai berikut.

$$\text{Rp } 3.800,00 = \text{Rp } 1.000,00 (3) + \text{Rp } 500,00 (1) + \text{Rp } 200,00 (1) + \text{Rp } 100,00 (1)$$

Jadi uang logam yang diperoleh konsumen sebanyak 6 uang logam yaitu uang Rp 1.000,00 sebanyak 3 uang logam, Rp 500,00 sebanyak 1 uang logam, Rp 200,00 sebanyak 1 uang logam, dan Rp 100,00 sebanyak 1 uang logam.

Dari analisis diatas diperoleh bahwa algoritma *Greedy* dapat memberikan solusi optimal pada program *vending machine*.

4. SIMPULAN

Umumnya *vending machine* tidak memberikan uang kembalian pada saat transaksi, hal ini tentu akan merugikan pihak konsumen. Dalam tulisan ini, didesain *vending machine* yang dapat memberikan uang kembalian sehingga harga minuman atau makanan sesuai dengan harga pasaran. Disini algoritma *Greedy* diterapkan sebagai salah satu cara untuk menyelesaikan permasalahan uang kembalian dengan memunculkan uang logam seminimal mungkin. Penulis menyarankan agar *vending machine* yang memberikan uang kembalian dengan kemunculan uang logam sesedikit mungkin dapat diterapkan pada *vending machine* di Indonesia.

5. REFERENSI

- [1] Pradeepa, P, dkk. 2013. Design and Implementation of Vending Machine Using Verilog HDL. *International Journal of Advanced Engineering Technology*: 51.
- [2] Rentika, Rika, dkk. 2014. Teknik Pemodelan dan Simulasi Vending Machine. *Makalah USU*: 9 & 11.
- [3] Nugroho, Aryo. 2010. Penggunaan Algoritma Greedy dalam Aplikasi Vending Machine. *Jurnal STEI ITB*: 1–4.